

APPENDIX "A"

module notifier\_disable;

initial begin

  `ifdef dis\_clr\_rem

    force notifier\_test.flop1.notifier\_clr\_rem = 1'bx;

  `endif

  `ifdef dis\_clr\_rec

    force notifier\_test.flop1.notifier\_clr\_rec = 1'bx;

  `endif

  `ifdef dis\_pos\_sh

    force notifier\_test.flop1.notifier\_pos\_sh = 1'bx;

  `endif

  `ifdef dis\_neg\_sh

    force notifier\_test.flop1.notifier\_neg\_sh = 1'bx;

  `endif

  `ifdef dis\_cp\_low

    force notifier\_test.flop1.notifier\_cp\_low = 1'bx;

  `endif

```
`ifndef dis_clr_high
    force notifier_test.flop1.notifier_clr_high = 1'bx;
`endif

`ifndef dis_cp_high
    force notifier_test.flop1.notifier_cp_high = 1'bx;
`endif

end

endmodule
```

APPENDIX "B"

```
`timescale 1ps/1ps

module notifier_test;

reg CLR;
reg CP;
reg D;
integer i;

DFF_QZ_CLR_X2_LD50      flop1 (
                        .QZ(QZ),
                        .CLR(CLR), .CP(CP), .D(D)
                        );

initial begin
    for (i = 0; i <= 2; i = i + 1) begin
        #1000      CLR = 1'b1;
        #1000      D = 1'b1;
                   CP = 1'b0;
        #5000      CLR = 1'b0;
    // Invalid CP high pulse width - notifier7
        #1000      CP = 1'b1;
        #1000      CP = 1'b0;
```

// Valid CP Pulse width  
#5000 CP = 1'b1;  
#5000 CP = 1'b0;  
// Invalid CLR high pulse width - notifier6  
#5000 CLR = 1'b1;  
#1000 CLR = 1'b0;  
// Invalid CLR deassertion to CP - notifier2  
#5000 CLR = 1'b1;  
#5000 CLR = 1'b0;  
#300 CP = 1'b1;  
// Valid CLR assertion  
#5000 CP = 1'b0;  
#5000 CLR = 1'b1;  
// Invalid CP to deassertion of CLR - notifier1  
#5000 CP = 1'b1;  
#300 CLR = 1'b0;  
// Valid CLR assertion  
#5000 CLR = 1'b1;  
// Invalid CP low pulse width - notifier5  
#5000 CLR = 1'b0;  
#5000 CP = 1'b0;  
#2000 CP = 1'b1;  
// Valid CP pulse width  
#5000 CP = 1'b0;  
#5000 CP = 1'b1;  
// Invalid setup/hold time negedge D to CP - notifier3

#5000 CP = 1'b0;  
#5000 D = 1'b0;  
#1000 CP = 1'b1;  
// Valid setup/hold time D to CP  
#5000 CP = 1'b0;  
#5000 CP = 1'b1;  
// Invalid setup/hold time posedge D to CP - notifier2  
#5000 CP = 1'b0;  
#5000 D = 1'b1;  
#1000 CP = 1'b1;  
`ifdef add\_simul  
// Valid setup/hold time D to CP  
#5000 CP = 1'b0;  
#5000 CP = 1'b1;  
// Invalid CP low time and D setup time  
#5000 CP = 1'b0;  
#3000 D = 1'b0;  
#100 CP = 1'b1;  
`endif  
#10000;  
end // for  
end // initial  
  
initial begin  
\$recordfile("notifier\_test.trn");  
\$recordvars("depth=0");

DOCKET NO. P05062

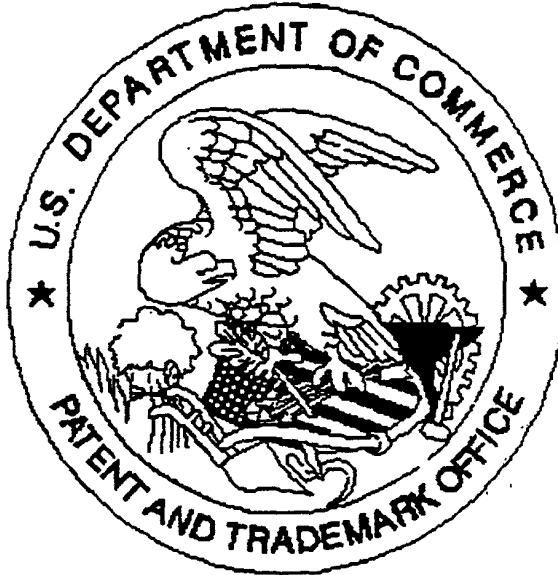
PATENT

end

endmodule

Patent - 6032500

United States Patent & Trademark Office  
Office of Initial Patent Examination -- Scanning Division



Application deficiencies found during scanning:

☐ Page(s) \_\_\_\_\_ of \_\_\_\_\_ were not present  
for scanning. (Document title)

☐ Page(s) \_\_\_\_\_ of \_\_\_\_\_ were not present  
for scanning. (Document title)

☐ *Scanned copy is best available.*

*Pages numbered 23-28 as part of specification  
are Appendices.*